

EncapReadme: Encap based bootstrap for bcfg2 and complete bcfg2 toolchain

This code is a method for getting bcfg2, including all dependencies, up and running on many platforms as quickly as possible, from source.

What you get

The end result is a self-extracting/self-installing bcfg2 client distribution that does a complete client install, which includes:

- the epkg encap package manager
- all software on which bcfg2 depends
- bcfg2 itself
- ostiary to kick off bcfg2 client runs remotely
- runit to run bcfg2 client as a periodic service with logging
- all with site-specific configuration parameters, set at build time in a single unified build-time configuration file, site-settings.conf
- optional install-time entry of bcfg2 and ostiary passwords, interactively or via environment variables

As well as:

- encap packages for software on which the bcfg2 server functionality depends (glib, gamin, and cheetah)
- encaps of optional documentation packages

Internet resources

For a more general overview, see <http://www.bcfg2.org/wiki/EncapPackages>

You can obtain the latest version of the code from bcfg2 svn:

- `svn co https://svn.mcs.anl.gov/repos/bcfg/trunk/bcfg2/encap`

Operation notes

Using the encap package manager this code automatically downloads, builds and creates encap packages for bcfg2 and dependencies to `/usr/local/encap`, and installs symbolic links to these packages from `/usr/local` (the usual encap thing).

It attempts to be as self contained as possible; everything gets linked to under `/usr/local/lib/bcfg2`, except for bcfg2 itself and some dependent software, which is prefixed by `b2-` (`b2-openssl`, `b2-python` etc.).

To run the bcfg2 server, you also need to install gamin, which supports a subset of the platforms bcfg2 client will work on, including GNU/Linux (but first install glib, on which gamin depends). You also need to install the cheetah

templating system on the bcfg2 server if you wish to use the bcfg2 templating functionality.

Important differences from upstream sources

- In general, everything is under `/usr/local` instead of `/`
 - ♦ `/usr/local/etc/bcfg2.conf` is used instead of `/etc/bcfg2.conf`

Environment variables and Sentinel files

Before the initial make/gmake and before the client install, you can set some environment variables to control some behaviors:

- `DEST="<path>"` – Set where the final build output goes. Default is `./DIST`
- `REPLACE_CONFIG="yes"` – Unconditionally replace local configuration files for bcfg2 and ostiary with those included in the distribution. The old files are saved to `<filename>--<date>`.
- `LOC_BCFG2_PASSWD="<password>"`, `LOC_OST_PASSWD="<password>"` – Set the bcfg2 server and ostiaryd daemon passwords, to avoid being interactively prompted for them.

There are also some "sentinel files" (zero byte files that only indicate state) that you can create to control the operation of the install. This is mostly useful so that installs don't clobber local changes / changes made by bcfg2.

Sentinel file names:

- `.SENTINEL_SITE` – Indicates that the bcfg2 client has been previously installed.
- `.SENTINEL_BCFG2` – Indicates that the files have been modified by bcfg2 itself. (If you change any of the config files mentioned below via bcfg2, you'll want to put this sentinel file in the appropriate directory with bcfg2 as well).

If either of these files exist, the install will not overwrite the existing config files unless `REPLACE_CONFIG="yes"` is set.

Directory with sentinel file(s)	Covered config files
-----	-----
<code>/usr/local/etc</code>	<code>bcfg2.conf</code> , <code>ostiary.conf</code>
<code>/usr/local/etc/default/bcfg2-client</code>	<code>env/RUN_INTERVAL_SECONDS</code> , <code>env/OPTIONS</code>
<code>/usr/local/etc/default/bcfg2-server</code>	<code>env/OPTIONS</code>
<code>/usr/local/sbin</code>	<code>ost-bcfg2.sh</code>

About runit integration

In order to avoid a lot of platform/distribution-specific code, the encap bcfg2 distribution includes and uses [<http://smarden.org/runit/useinit.html>] runit] instead of init scripts and cron.

The bcfg2 client (.run) distribution uses runit to run ostiary, and to run the bcfg2 client periodically.

On the server, edit `/usr/local/etc/default/bcfg2-server/env/OPTIONS` to include the options you want to start up the bcfg2 server with, and then do

```
ln -s /usr/local/etc/sv/bcfg2-server /usr/local/var/service/
```

to enable the service.

You can use `sv status /usr/local/var/service/bcfg2-server` to see the status, and `rm /usr/local/var/service/bcfg2-server` to remove it.

Logs for all runit services are under `/usr/local/var/svlogd`.

About ostiary integration

In order to enable the remote kickoff of bcfg2 client runs, the bcfg2 client distribution includes ostiary, a simple, very security-paranoid daemon that runs a script with fixed arguments based on a password hash it receives.

The following actions are available via ostiary; you can add more by editing `/usr/local/etc/ostiary.cfg`. The `<password>` is a value you set during compile-time or (preferably) `.run` file install time.

- `<password>-bcfg2-dvqn` : Run `bcfg2-client -d -v -q -n`
- `<password>-bcfg2-dvn` : Run `bcfg2-client -d -v -n`
- `<password>-bcfg2-dvq` : Run `bcfg2-client -d -v -q`
- `<password>-bcfg2-dv` : Run `bcfg2-client -d -v`
- `<password>-bcfg2-vq` : Run `bcfg2-client -v -q`
- `<password>-bcfg2-v` : Run `bcfg2-client -v`
- `<password>-bcfg2-restart` : Restart the `bcfg2-client` runit service

There are plans for the future for a bcfg2 plugin that will set per-machine passwords after the initial install, however as with cfengine the worst that someone can do if they find your password is to bring your host into a cleaner state.

To execute one of these actions, you use the `ostclient` command, i.e.: `ostclient -a <address> -p <port>` where `<address>` is the address of the machine you want to run the bcfg2 client on, and `<port>` is the ostiary port number you set during the INSTALL procedure. You will then be prompted to `Enter command secret:`, at which point you will enter one of the above-listed values, such as `<password>-bcfg2-dvqn` (the command to run and the password are integrated into the same string).

Logs of bcfg2-client runs kicked off via ostiary are in `/usr/local/var/svlogd/bcfg2-client-ostiary`

Supported Platforms

Below is a table of platforms that have been successfully bootstrapped using this code.

OS	Vendor	Version	Arch	GCC	By	Bcfg2
AIX	IBM	5.2	POWER	3.3.2	dc	0.8.5
AIX	IBM	5.3	POWER	4.1.1	dc	0.8.5
GNU/Linux	Debian	Sarge	i386	3.3.5	dc	0.8.5
GNU/Linux	Debian	Etch	x86_64	4.1.2	dc	0.9.2
GNU/Linux	Debian	Sid	i386	4.1.2	dc	0.8.5
GNU/Linux	SuSE	SLES8	i386	3.2.2	dc	0.8.5
GNU/Linux	SuSE	SLES10	i386	4.1.0	dc	0.8.5
GNU/Linux	Ubuntu	Dapper	i386	4.0.3	dc	0.8.5

dc: "Daniel Clark" <<mailto:dclark@member.fsf.org>>

If you bootstrap a platform not listed above, please add a comment to:

- <http://trac.mcs.anl.gov/projects/bcfg2/ticket/74>

so that platform can be added to the list.

If you modified any of the files in this package to be able to bootstrap the new platform, please include either diffs or a tarball of your modified version in a new ticket so your changes can be incorporated into a new release.

Any other notes, such as where you got the GNU binaries or any issues people should be aware of, would also be appreciated.

You may want to scan all of the bootstrapped binaries and libraries with `ldd` (or equivalent) to make sure there are no dependencies on libraries other than those included with the base operating system and the libraries built as part of the bootstrap process.

Encap profile (.ep) documentation

Note that the doc for the encap profile format is in ``man 5 encap_profile``.

Next steps

1. Build and install; see [INSTALL](#)
2. Set up your server and clients; see [HOWTO](#)

Documentation Version

- This is a copy of: \$Id: README 2856 2007-02-20 03:08:47Z dclark \$
- Most recent version: <http://www.bcfg2.org/browser/trunk/bcfg2/encap/README>